

Rで学ぶプログラミングの基礎の基礎

(7) 繰り返し文



プログラミングについて

- ▶ プログラミングとは
 - ▶ 人間がコンピュータに命令をすること
 - ▶ R の場合は「ユーザーが R のコマンドをひとつひとつ記述する」作業のこと ⇒ 関数定義！
- ▶ R におけるプログラミングのための道具は・・・
 - ▶ 変数，ベクトル，関数定義
 - ▶ 条件分岐 (if)
 - ▶ **くり返し (for)**





本日のメニュー

1. **for 文**
2. for 文 + if 文
3. while 文



くり返し〔for〕

- ある「処理」をくり返し実行する \Rightarrow for を用いる

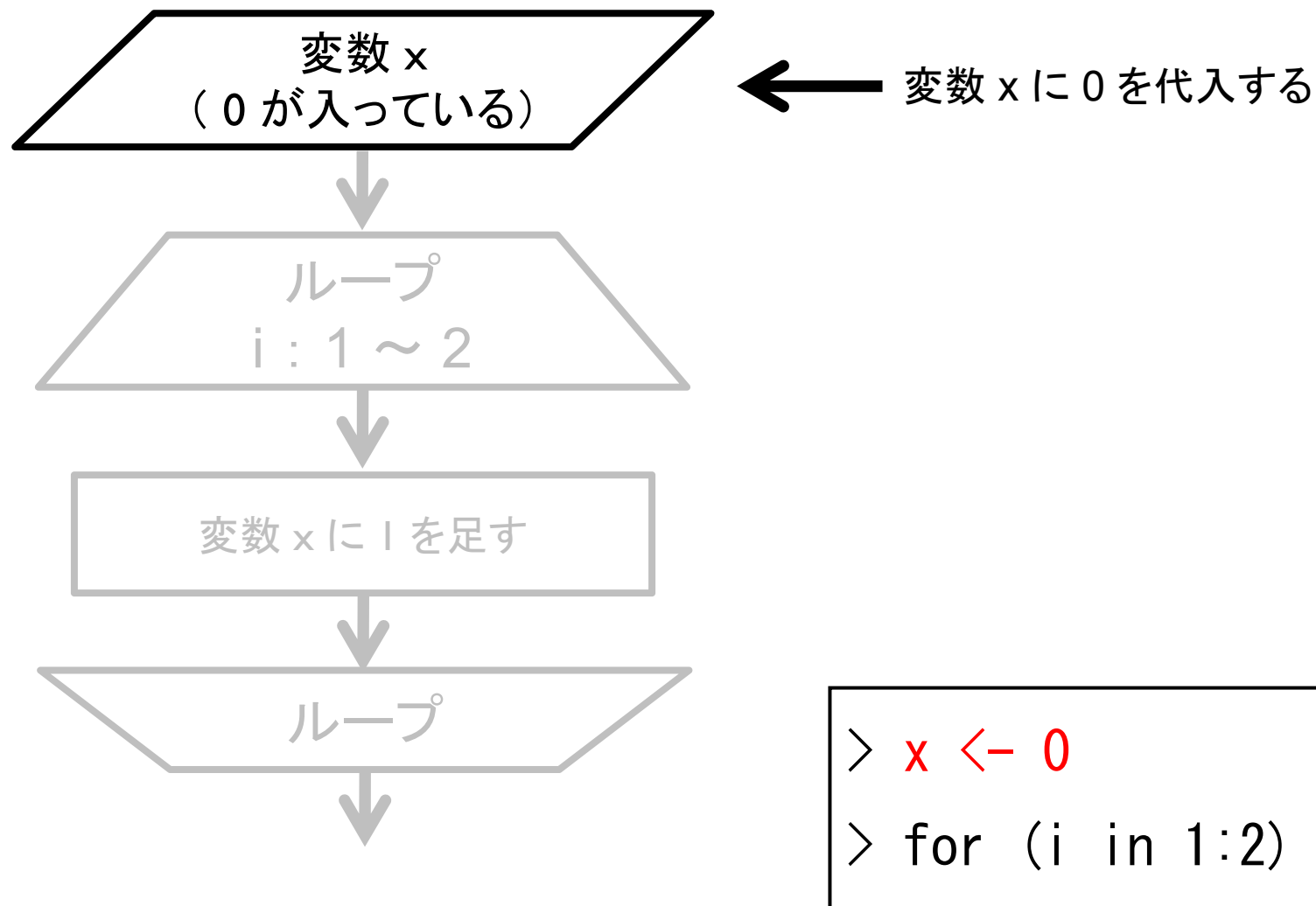
```
for (i in 1:くり返し数) { くり返し実行する処理 }
```

- 「処理」として「変数 x に 1 を足す」を 2 回くり返す例を挙げる

```
> x <- 0                                # x に 0 を代入
> for (i in 1:2) x <- x+1                # 「x に 1 を足す」を
                                         # 2 回くり返す
> x                                       # x の中身を確認
[1] 2
```

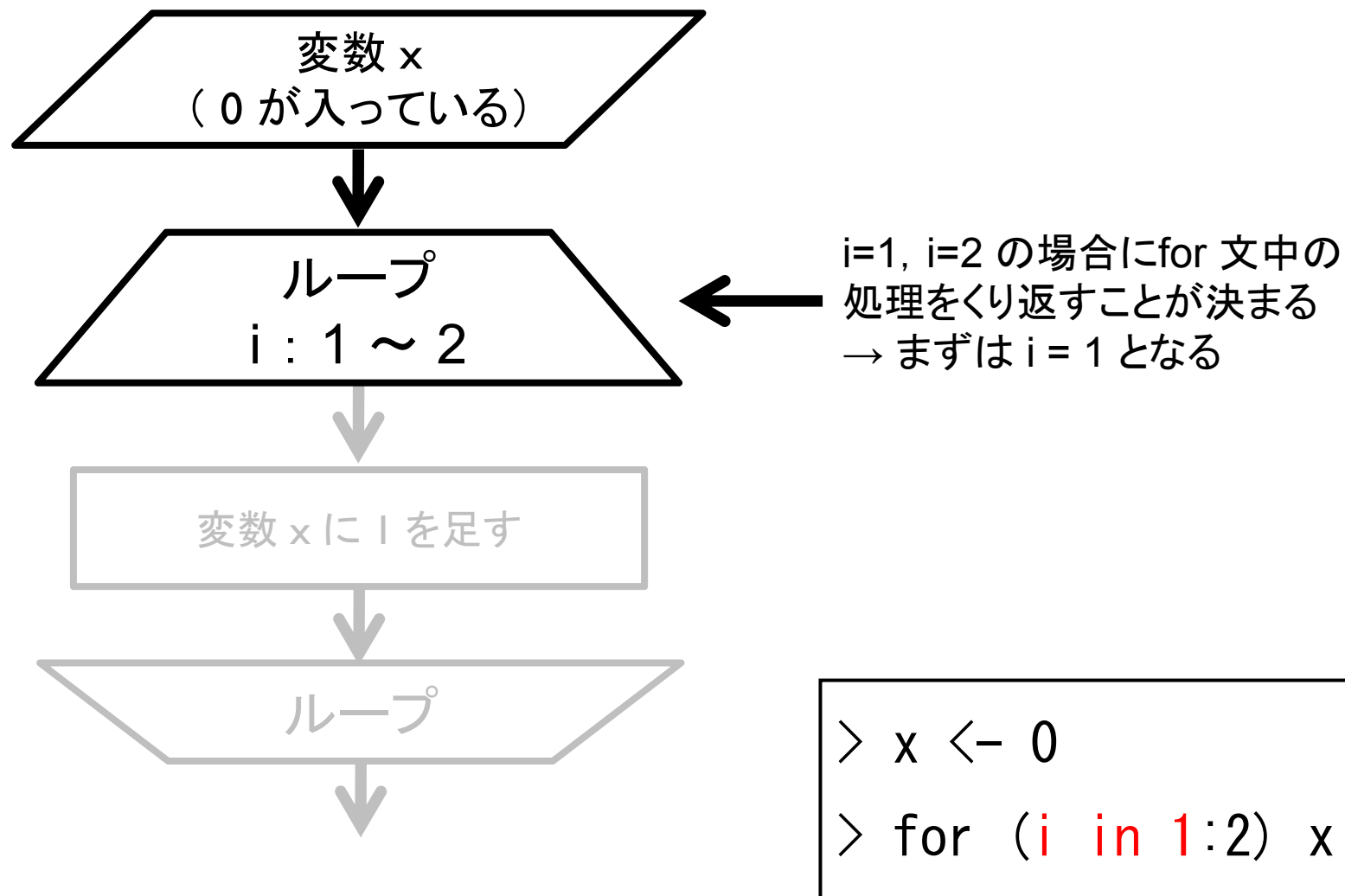


くり返し〔for〕



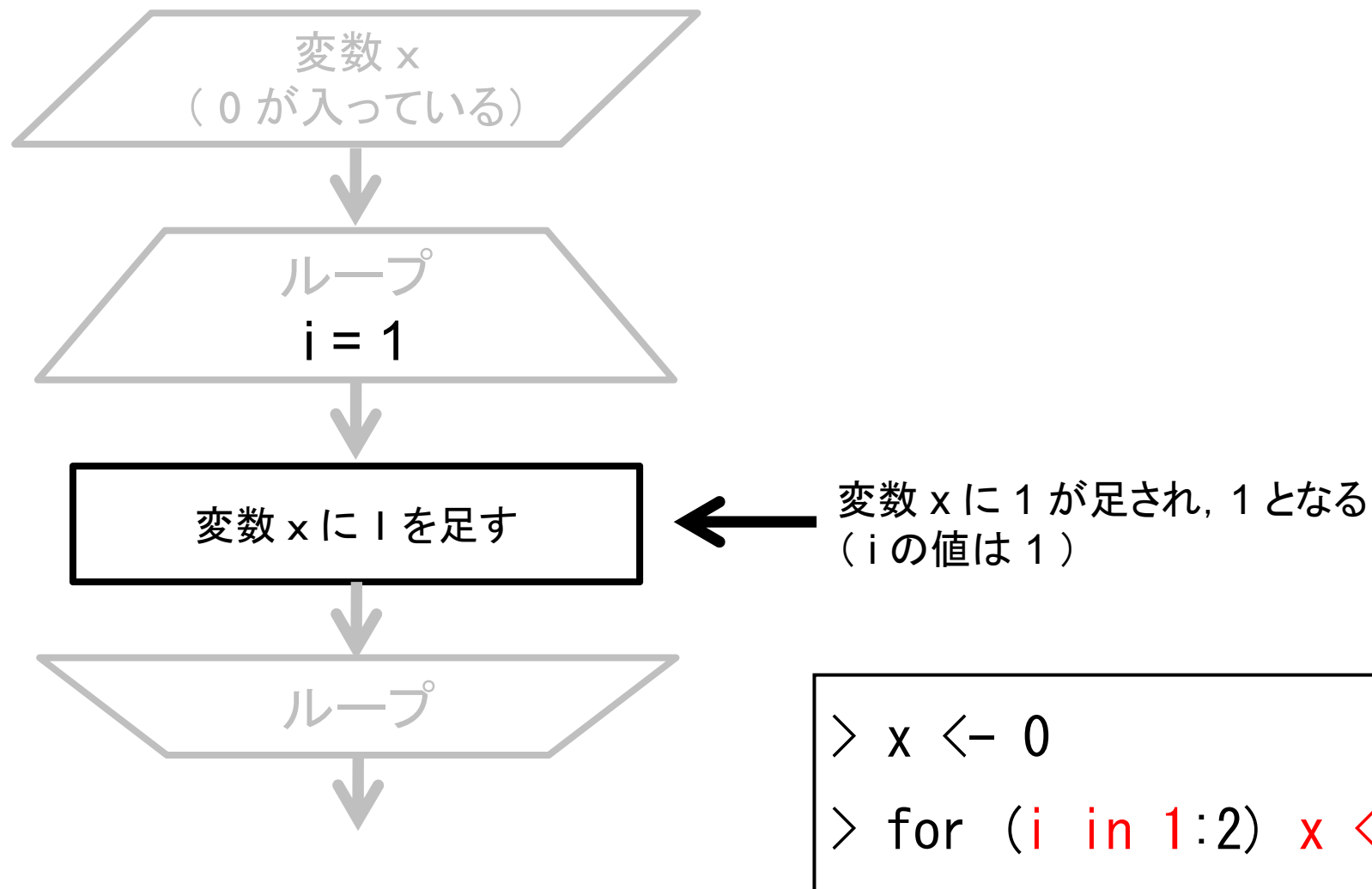


くり返し〔 for 〕



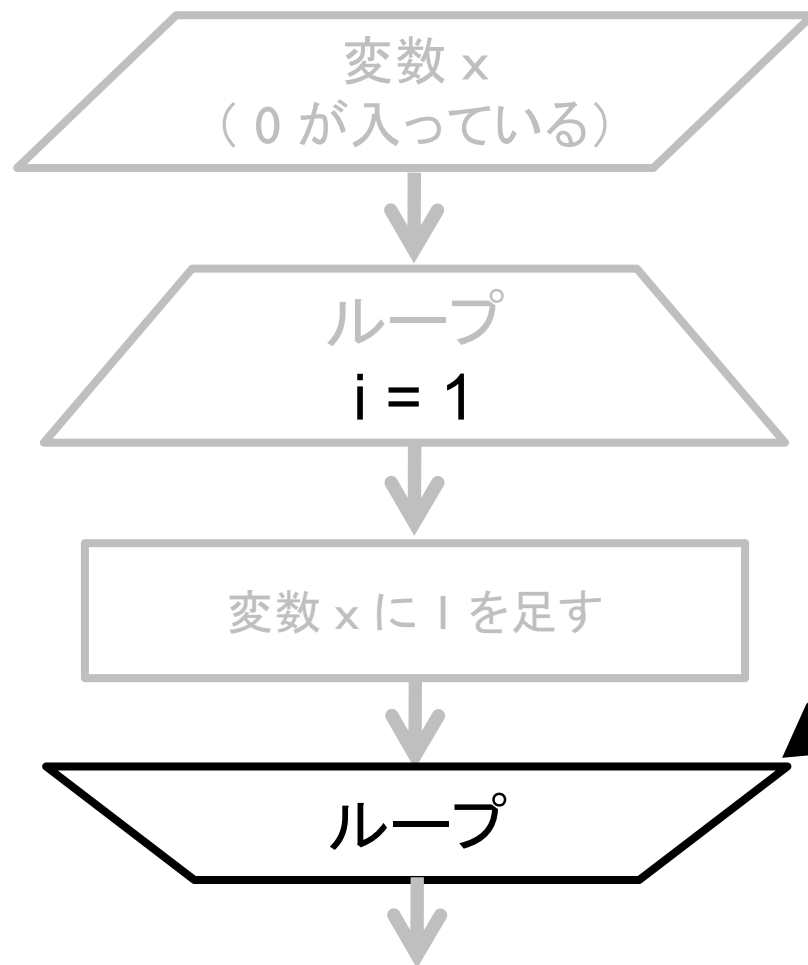


くり返し〔 for 〕





くり返し〔 for 〕

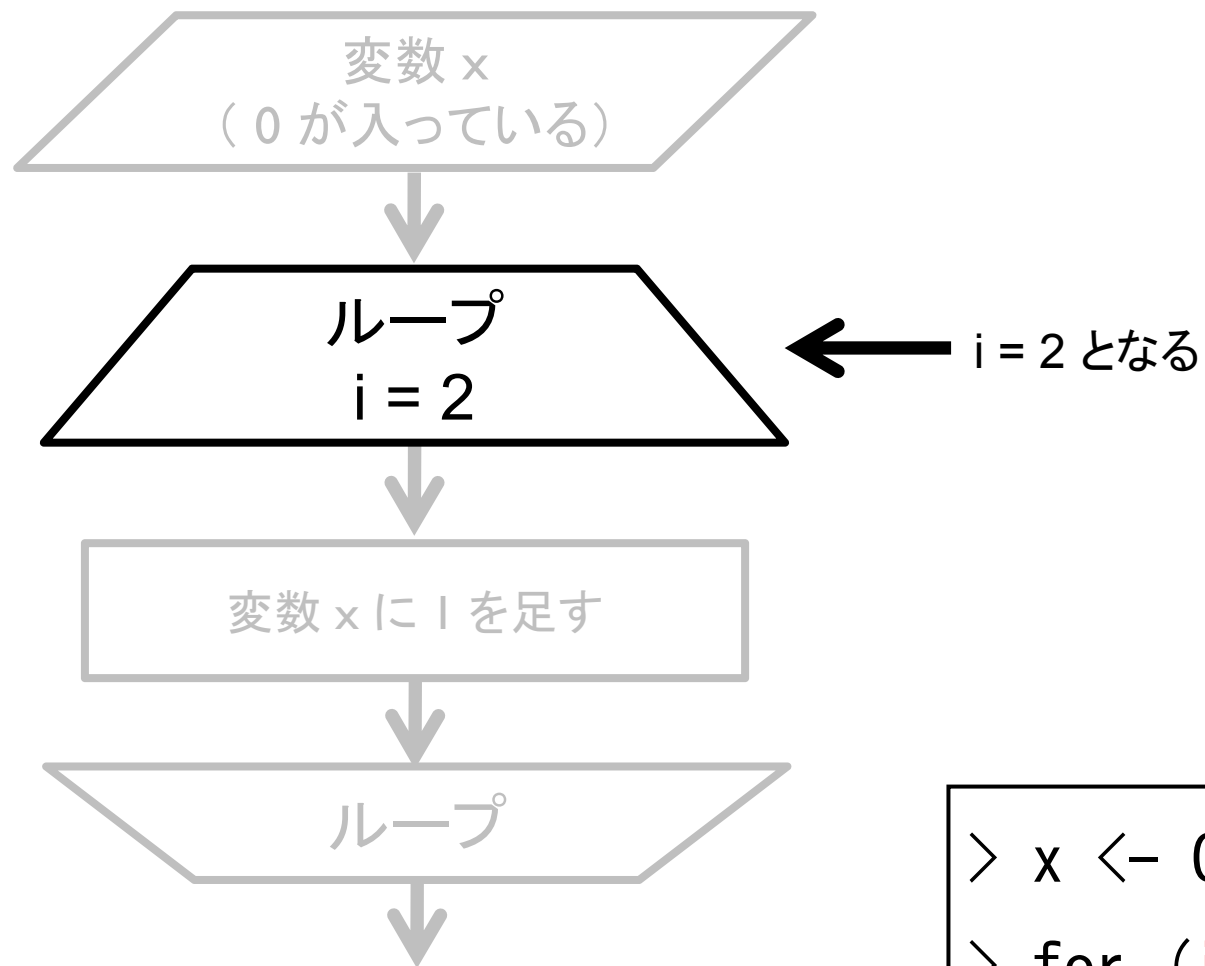


i=1 なので, ループは終了しない
→ ループの先頭に戻る

```
> x <- 0  
> for (i in 1:2) x <- x+1
```



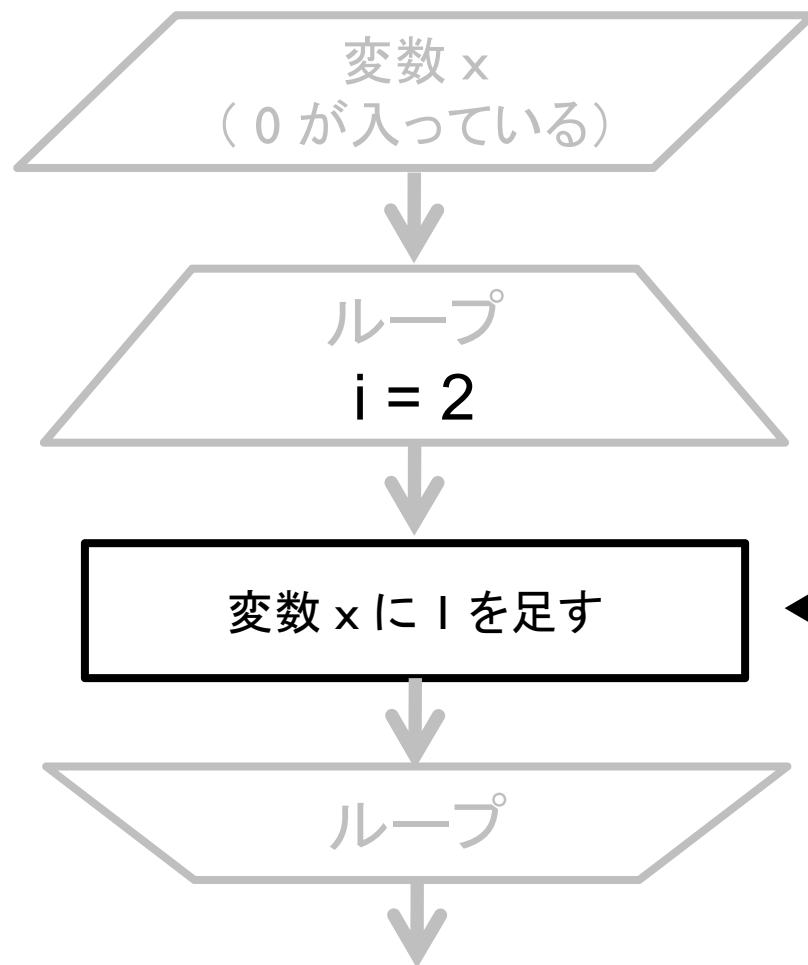

くり返し〔 for 〕



```
> x <- 0  
> for (i in 1:2) x <- x+1
```



くり返し〔 for 〕

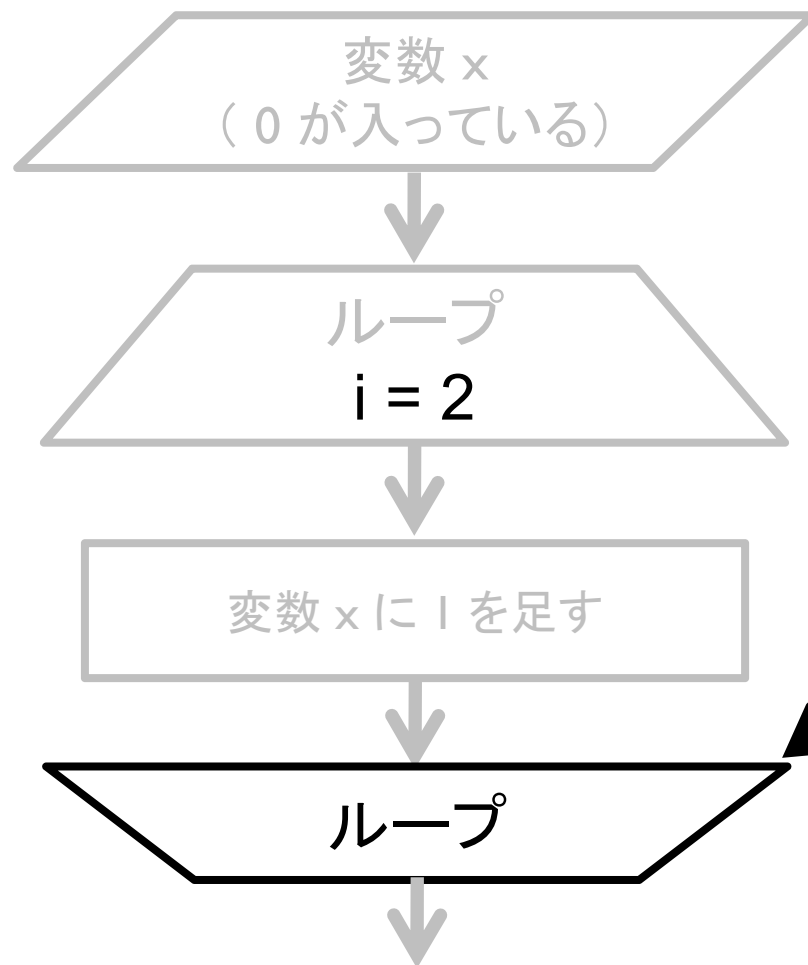


← 変数 x に 1 が足され, 2 となる
(i の値は 2)

```
> x <- 0  
> for (i in 1:2) x <- x+1
```



くり返し〔 for 〕



i=2 の処理が終了し, くり返し数の最後
なのでループが終了する

```
> x <- 0  
> for (i in 1:2) x <- x+1
```



〔 for 〕 の使用例

- ▶ 「変数 x に k を足す」を 5 回くり返す

```
> x <- 0                # x に 0 を代入
> for (k in 1:5) x <- x+k # x に k を足す
> x                      # x を表示
[1] 15
```

- ▶ 「ベクトル x に i をくっつける」を 5 回くり返す

```
> x <- c()              # 空のベクトルを用意
> for (i in 1:5) x <- c(x, i) # x に i をくっつける
> x                      # x を表示
[1] 1 2 3 4 5
```



本日のメニュー

1. for 文
2. **for 文 + if 文**
3. while 文



〔 for + if 〕 の使用例

- ▶ 「1 から x までの間の偶数を全て足し合わせる」という関数 `myeven()` を定義する

```
> myeven <- function(x) {  
+   a <- 0                                # a に 0 を代入  
+   for (i in 1:x) {  
+     if (i%%2 == 0) a <- a+i             # i を 2 で割った余り  
+                                           # が 0 なら i を足す  
+   }  
+   return(a)  
+ }  
  
> myeven(10)                             # 1~10 の偶数の和  
[1] 30
```



演習問題

1. 正の整数値 n を入れると 1 から n までの積 ($n!$) を返す関数 `myprod1()` を定義して下さい
2. 1 以上の奇数 n を入れると 1 から n のうち奇数だけを掛け算した二重階乗 ($n!!$) を返す関数 `myprod2()` を定義して下さい
→ 例えば, $n=7$ のときは「 $1 \times 3 \times 5 \times 7 = 105$ 」となります



本日のメニュー

1. for 文
2. for 文 + if 文
3. while 文



くり返し〔 while 〕

- ▶ while 文は、ある条件が成り立っている間（ TRUE の間）はずっと処理をくり返す
- ▶ 「ループ変数のリストの個数だけ処理を行うとくり返しが終了する」 for 文と異なり、「条件式が TRUE しかとり得ない」場合はプログラムが永遠に実行され続けて（暴走して）しまうので注意
- ▶ もし暴走した場合は、実行結果が返ってこず、Windows の砂時計のアイコンがクルクル回ります。その時は [Esc] キーを押すことで処理を止めることが出来る

```
while ( 条件式 ) { くり返し実行する処理 }
```



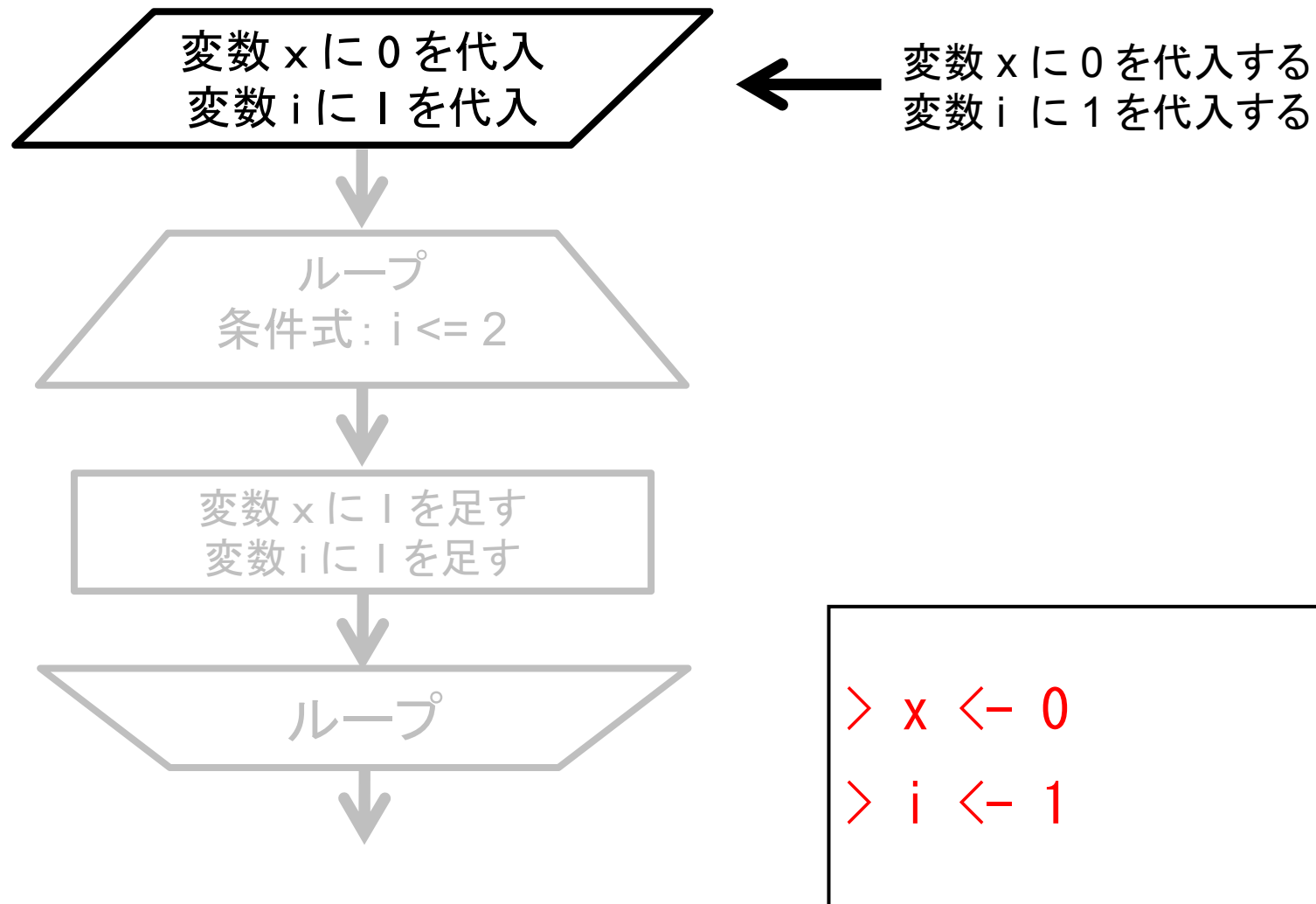
くり返し [while]

- ▶ 「変数 x に 1 を足す」を 2 回くり返す例を挙げる

```
> x <- 0
> i <- 1          # くり返しを制御するための変数
> while (i <= 2) {
+   x <- x + 1
+   i <- i + 1    # くり返し回数をカウント
+ }
> x
[1] 2
```

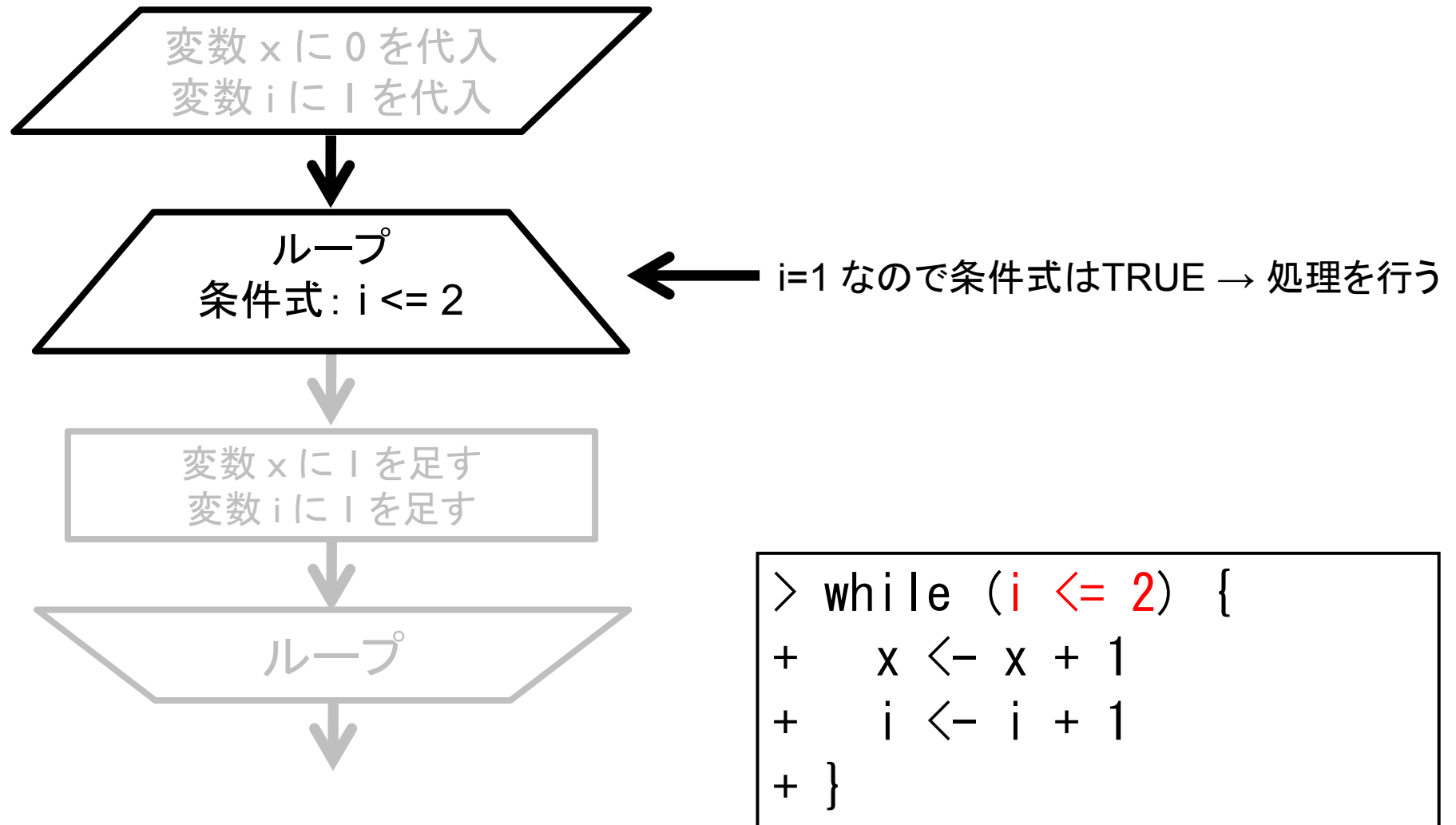


くり返し [while]



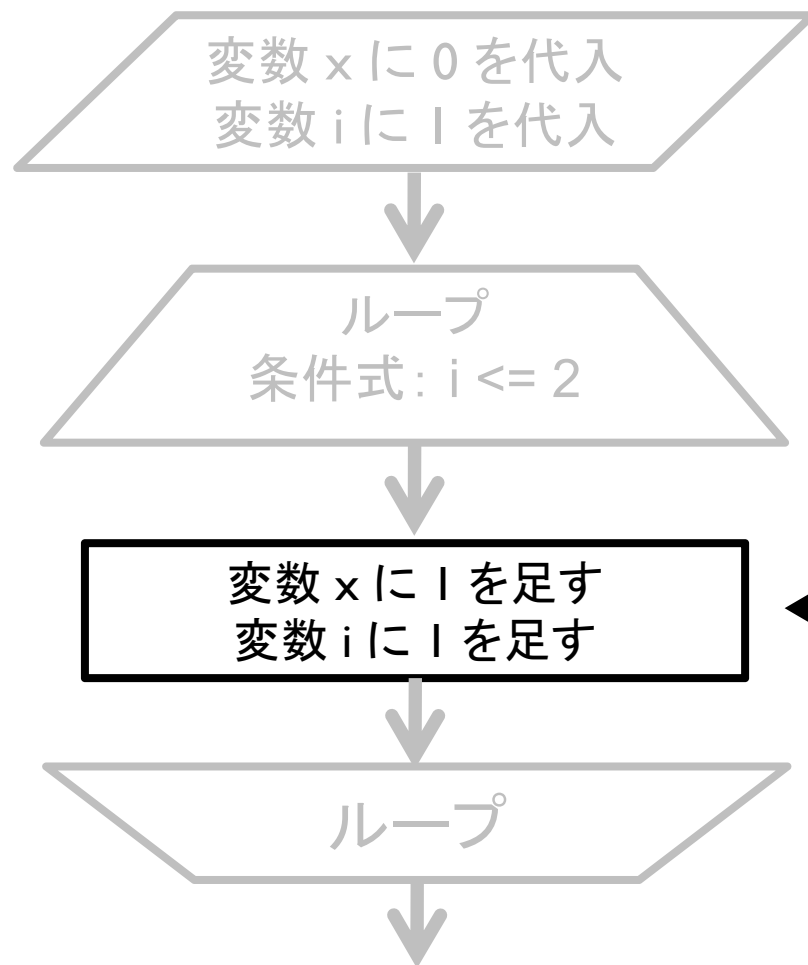


くり返し〔 while 〕





くり返し〔 while 〕

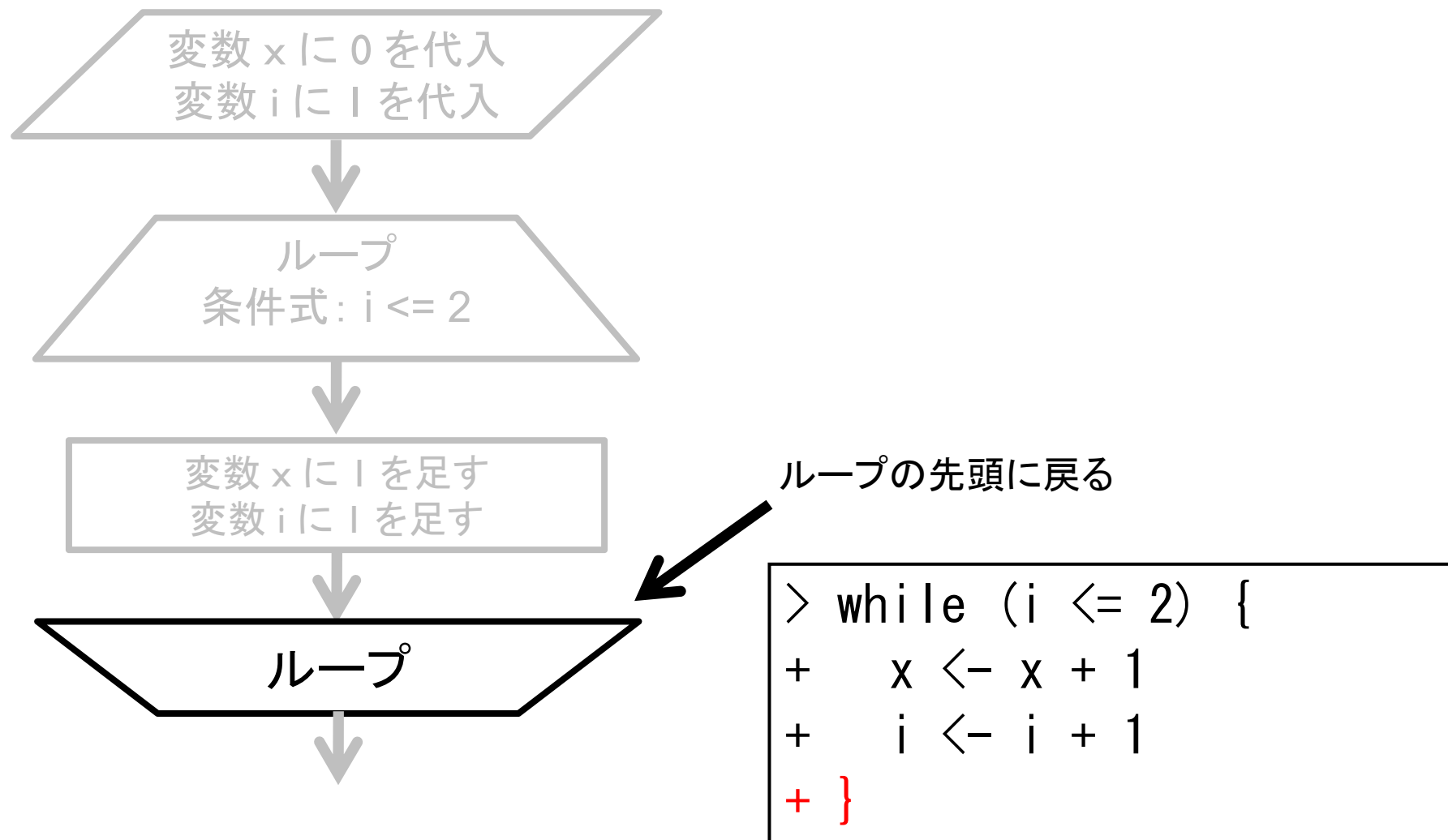


変数 x に 1 が足され, 1 となる
変数 i に 1 が足され, 2 となる

```
> while (i <= 2) {  
+   x <- x + 1  
+   i <- i + 1  
+ }
```

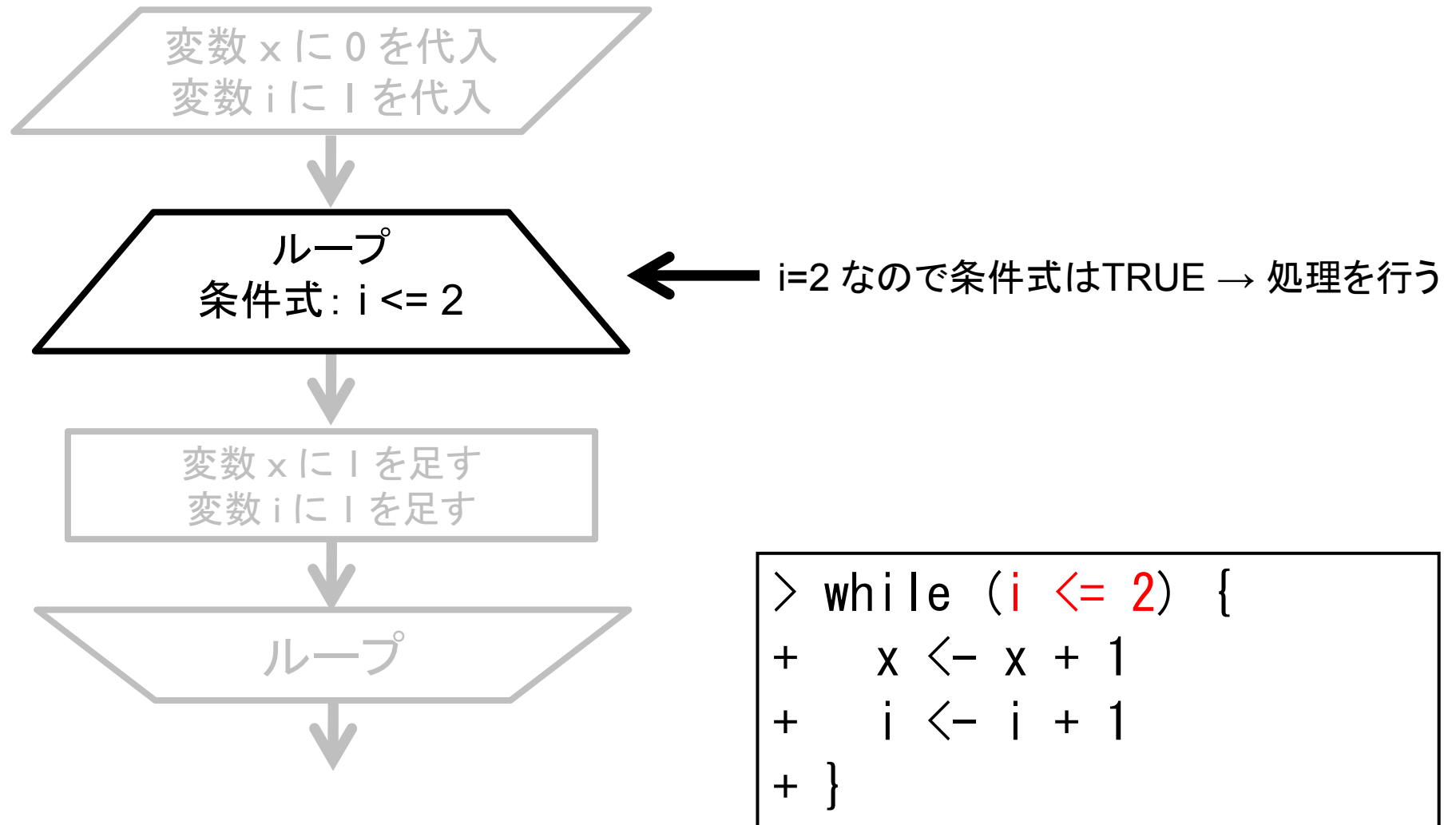


くり返し〔 while 〕



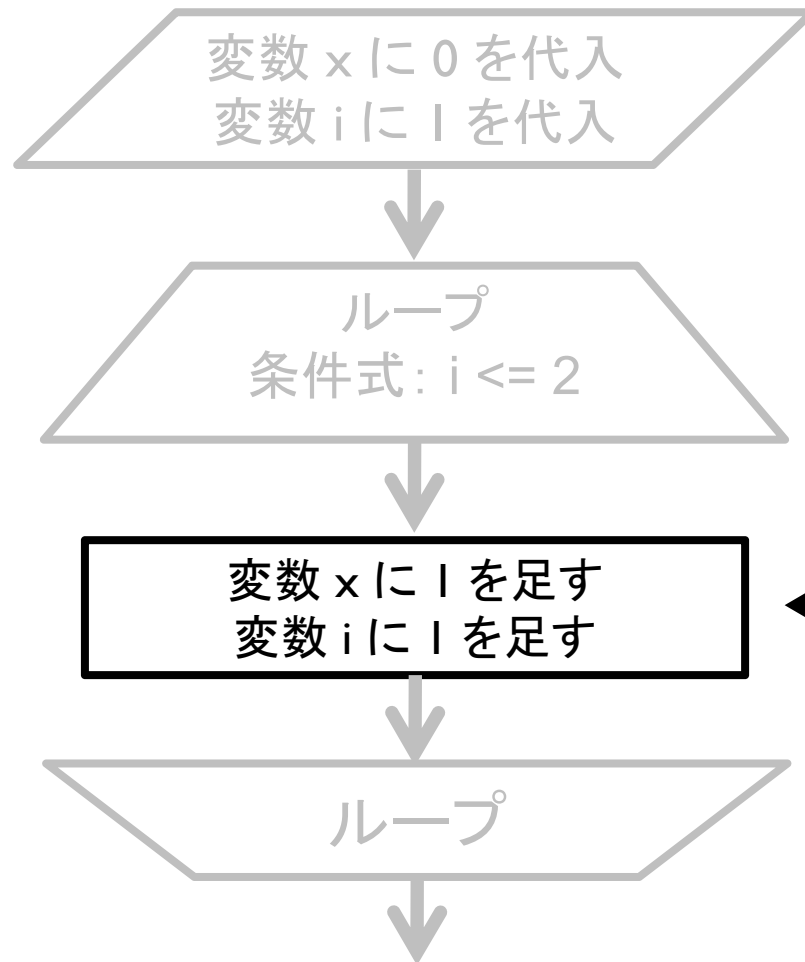


くり返し〔 while 〕





くり返し [while]

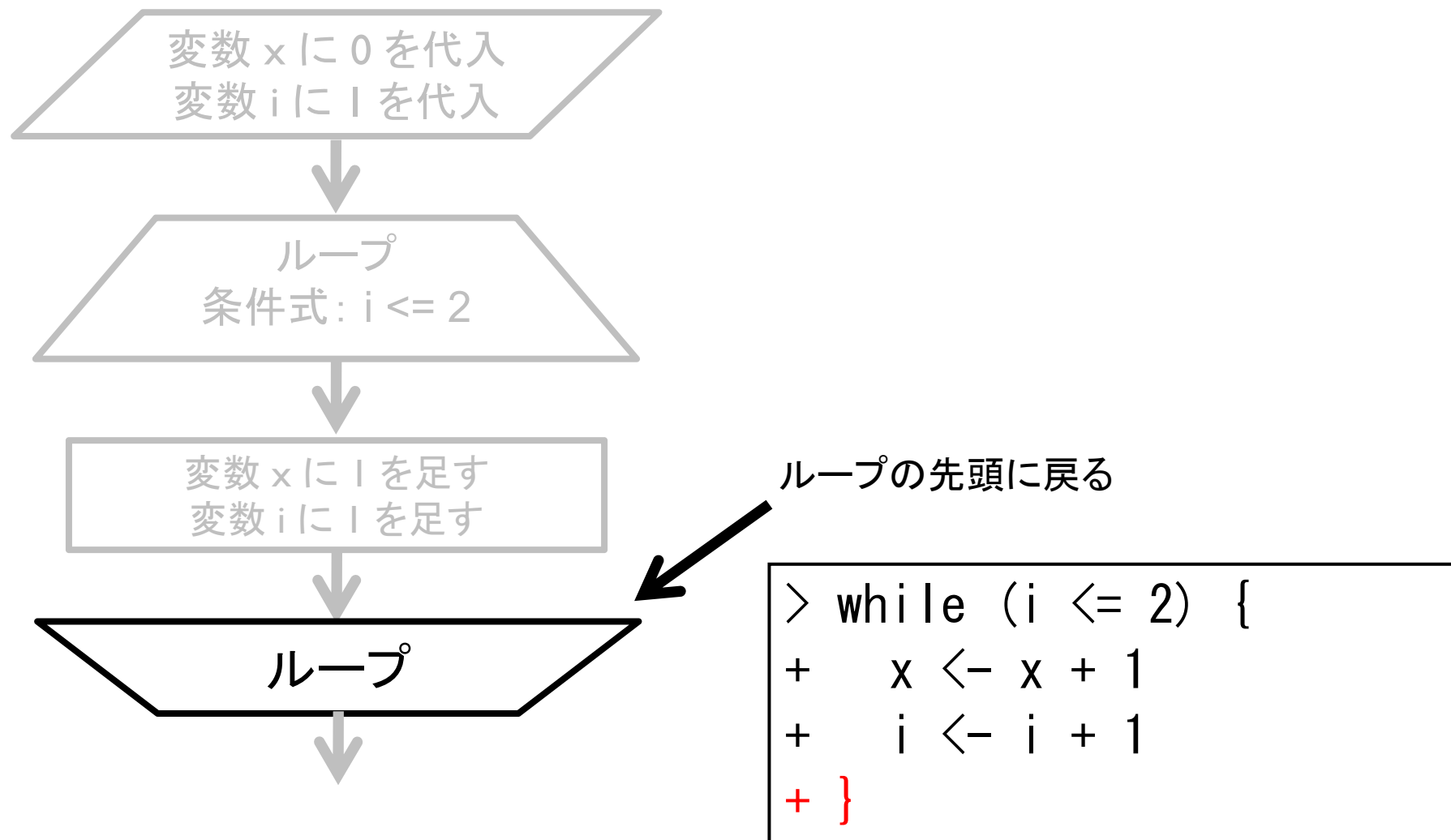


変数 x に 1 が足され, 2 となる
変数 i に 1 が足され, 3 となる

```
> while (i <= 2) {  
+   x <- x + 1  
+   i <- i + 1  
+ }
```

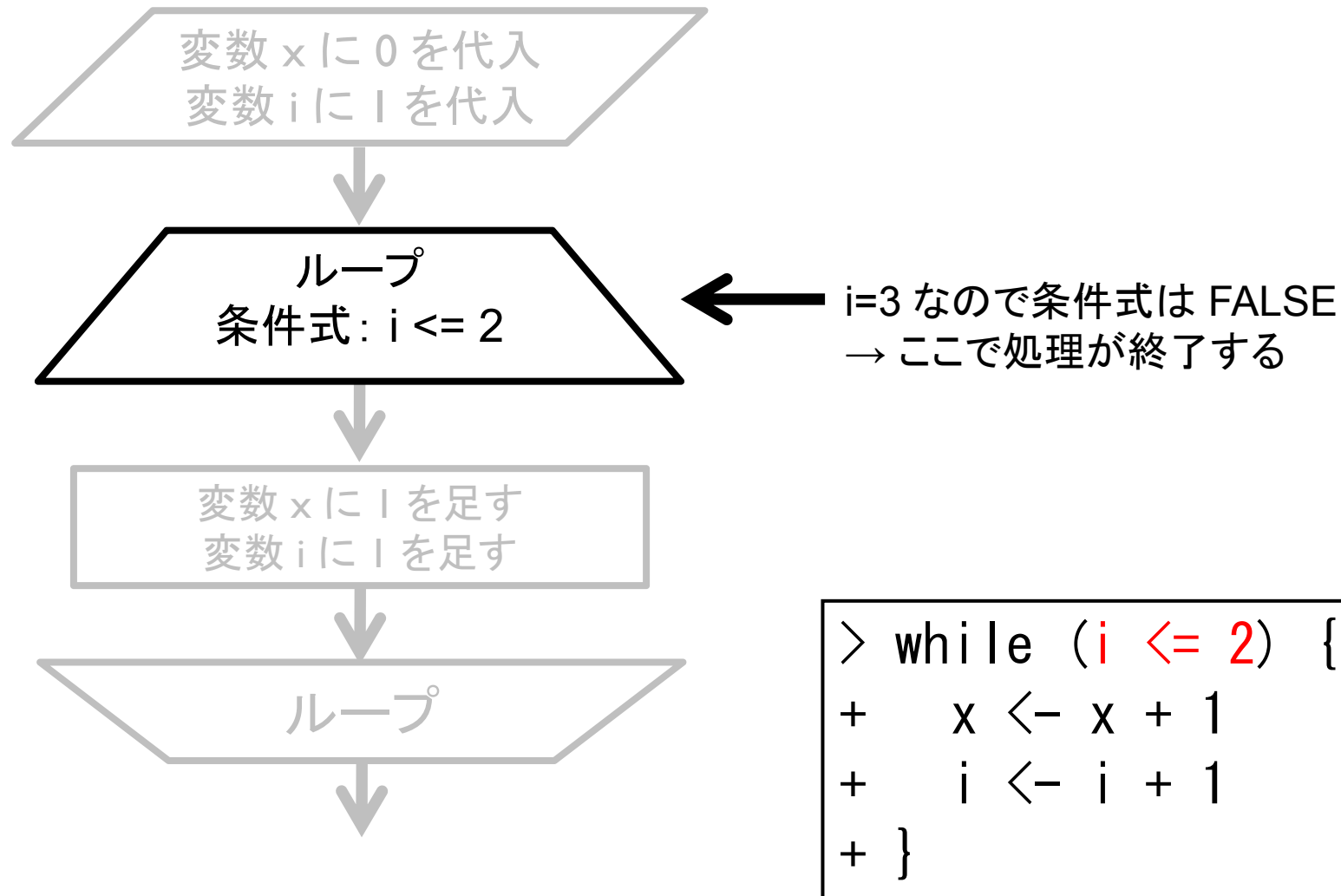



くり返し [while]





くり返し〔 while 〕





〔 while 〕 の使用例

- ▶ **while** 文は「くり返す回数が明確でない」ときに威力を発揮する
- ▶ 以下の関数 **myloop()** は実行すると「Y 又は N を入力してください：」と表示され，コンソール画面上で Y または N を入力するまで「Y 又は N を入力してください：」のメッセージが表示され続ける

```
> myloop <- function() {  
+   flag <- 0  
+   while (flag == 0) {  
+     x <- readline("Y又はNを入力してください：")  
+     if ( any(c("Y", "y") %in% x) ) {  
+       flag <- 1  
+       print("Yes.")  
+     }  
+     else if ( any(c("N", "n") %in% x) ) {  
+       flag <- 1  
+       print("No.")  
+     }  
+   }  
+ }
```



〔 while 〕 の使用例

- ▶ 「コンソール画面上で Y または N を入力するまで」ということはユーザーが何回誤入力をするかが分からない
→ このような場合に while 文は重宝する

```
> my loop ()
```

```
Y又はNを入力してください : a
```

```
Y又はNを入力してください : (何も入力せずに [Enter] )
```

```
Y又はNを入力してください : N
```

```
[1] "No."
```





本日のメニュー

1. for 文
2. for 文 + if 文
3. while 文



このスライドの内容が書籍になっています

- ▶ 舟尾 暢男 『R で学ぶプログラミングの基礎の基礎（カットシステム）』
「R と RStudio のインストール」「R の基礎」「変数とベクトル」
「種々のベクトル」「関数の作成」「条件分岐」「くり返し」
「複雑なくり返し」「グラフ作成の概要」「問題集」「簡単なゲーム作成」



Rで学ぶプログラミングの基礎の基礎

終